

# Weekly Report

Period: 18/06/17 - 25/06/17

Reporter: 高翔

## 1 Last Week

### 1. 把 3D 模型导入到 AR 环境中

Three.js 提供了 3D 模型的 Loader, 支持了许多格式的 3D 模型导入, 包括 \*.obj、\*.sea、\*.3mf、\*.amf、\*.sea、\*.pmd、\*.json 等。我在 <http://graphics.cs.williams.edu/data/meshes.xml> 找到了一个经典的人头模型, 将其导入到了 AR 中, 然后随着上周实现的人脸追踪得出的人脸坐标进行移动。

### 2. 在手机上跑通

以上流程都在电脑上没问题, 但是手机上要获取手机摄像头的 stream 一直不成功, 所以手机上没跑通。之前浏览器中获取手机摄像头数据用的是 navigator.getUserMedia 但这个 API 已经被废弃了, 新的 API 是 navigator.MediaDevices.getUserMedia, 但不能用。我在多个手机上进行了测试, 所以应该是 API 的问题, 这周我会再找找有没有什么其他办法获取摄像头数据。

### 3. 人脸识别 (recongnition, 之前实现的是 detection) 的进展

我分析了 Tracking.js 的源码, 发现它是拿到图片的每个像素之后 (<https://github.com/eduardolundgren/tracking.js/blob/master/src/tracking.js#L150>, <https://github.com/eduardolundgren/tracking.js/blob/master/src/tracking.js#L256>), 传给了指定的 tracker (<https://github.com/eduardolundgren/tracking.js/blob/master/src/tracking.js#L185>), 然后再加载特定的分类器, 最后把所有像素传给分类器进行了检测 (<https://github.com/eduardolundgren/tracking.js/blob/master/src/detection/ViolaJones.js#L44>)。

分类器中的数组数据是根据 OpenCV 训练好的数据转化而来 (<https://github.com/eduardolundgren/tracking.js/issues/92>, [https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\\_frontalface\\_alt2.xml#L55](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_alt2.xml#L55))

具体的 Haar-like 特征算法参考 <http://blog.csdn.net/xizero00/article/details/46929261>

不过 Tracking.js 只能实现 detection 不能实现 recongnition。要想识别谁是谁, 还得经过更复杂的过程:

- (a) 一种方法是用 OpenCV 是识别, OpenCV 应该有成熟的识别算法, 只要把前端获取的数据传到后端就可以了。
- (b) 第二种方法是用深度学习进行训练。可以用 ConvNetJS (<https://github.com/karpathy/convnetjs>) 在线训练, 但是速度估计会超级慢。也可以用 keras 预先训练好, 然后在浏览器中把训练好的网络模型加载进来, 用 (<https://github.com/transcranial/keras-js>) 加载。当然 TensorFlow 也可以训练, 然后通过 (<https://github.com/npinsker/tf2convnet>) 加载网络模型。

### 4. 新加坡可视化课程

这周主要和同学一起 code review, 并安排了下周的任务。目前和我做的工作, 可以复用, 到后来可能就分开了。

## 2 Next Week

- 1. 把人脸追踪到模型渲染都在手机上跑通
- 2. 试试将体绘制数据直接在浏览器中渲染
- 3. Chinavis 论文修改